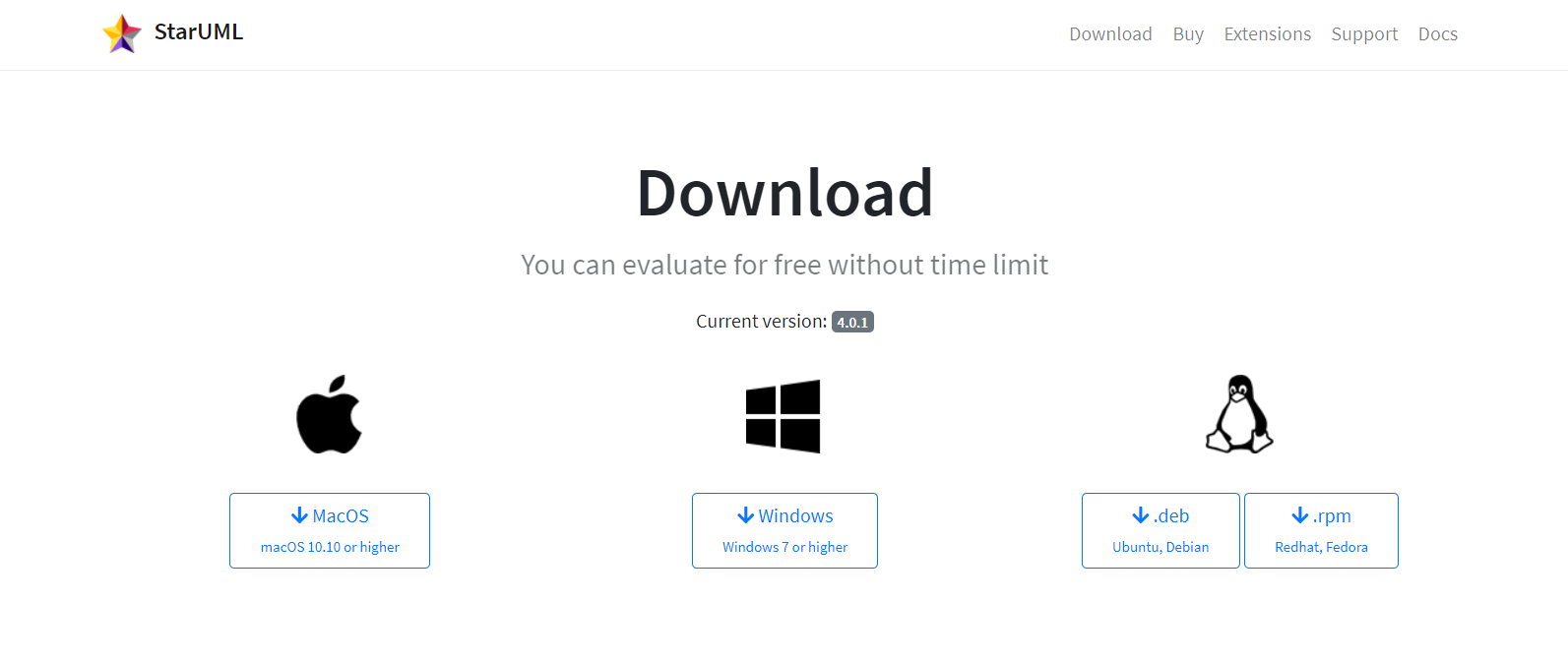
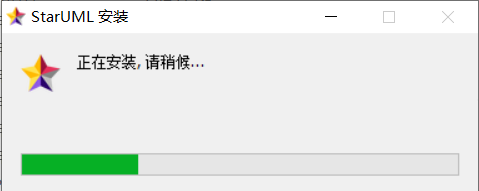
# 3.2 StarUML的安装和配置

### 3.2.1 StarUML的安装

（1）进入StarUML的官网<https://staruml.io/download> 下载最新版4.0.1安装包



（2）启动安装包，安装StarUML。安装过程为自动默认安装。

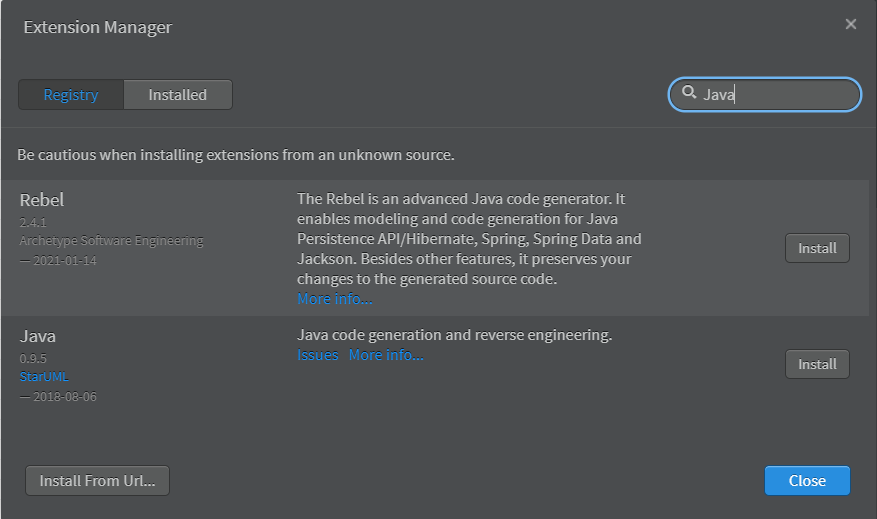


### 3.2.2 StarUML的配置

（1）为了能与面向对象的程序设计软件相关联，实现双向工程，需要在StarUML中配置Java插件

打开StarUML，通过菜单栏Tools下的Extension Manager下载插件

在搜索框中搜索Java，并安装插件

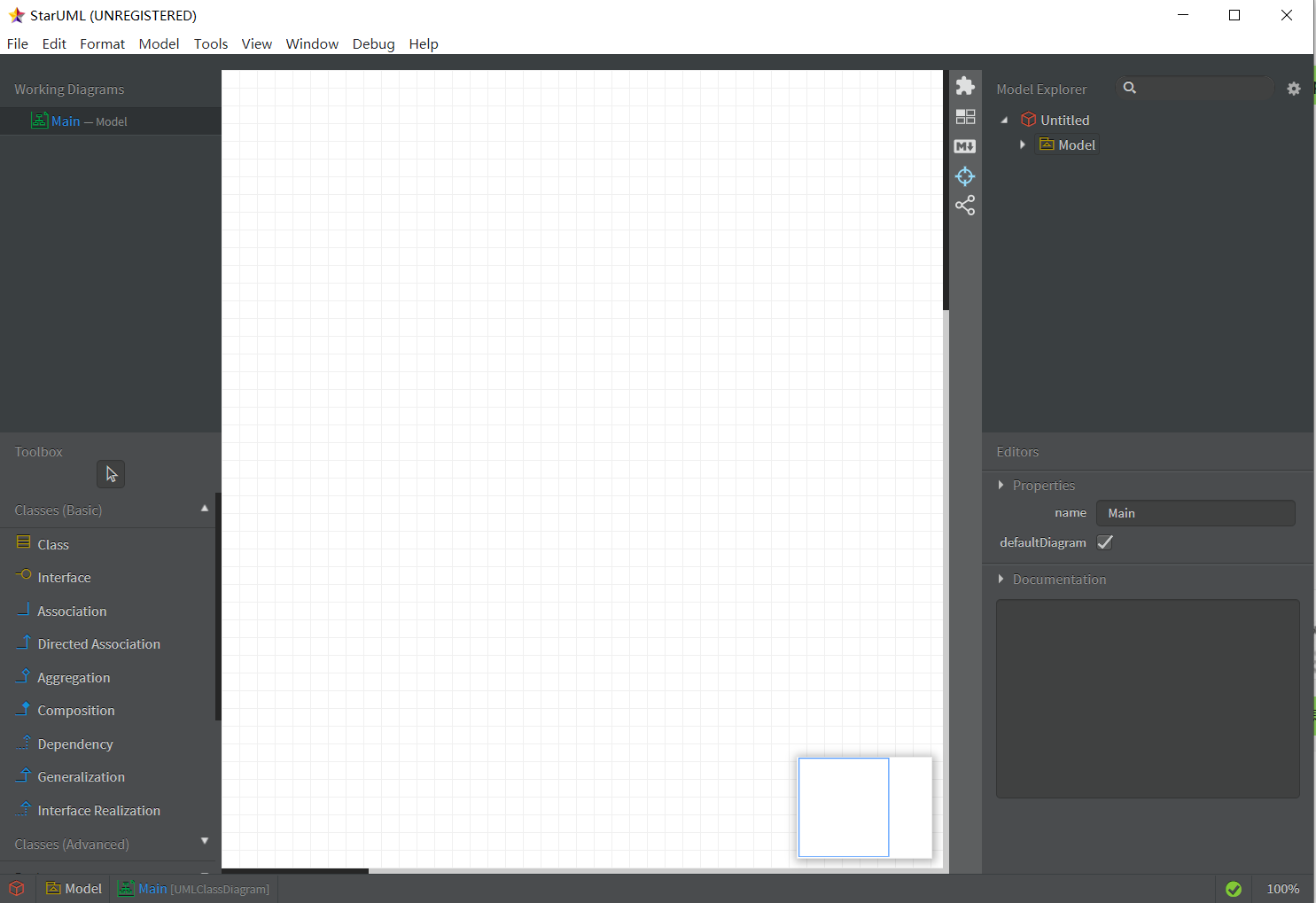


# 3.3 使用StarUML建模

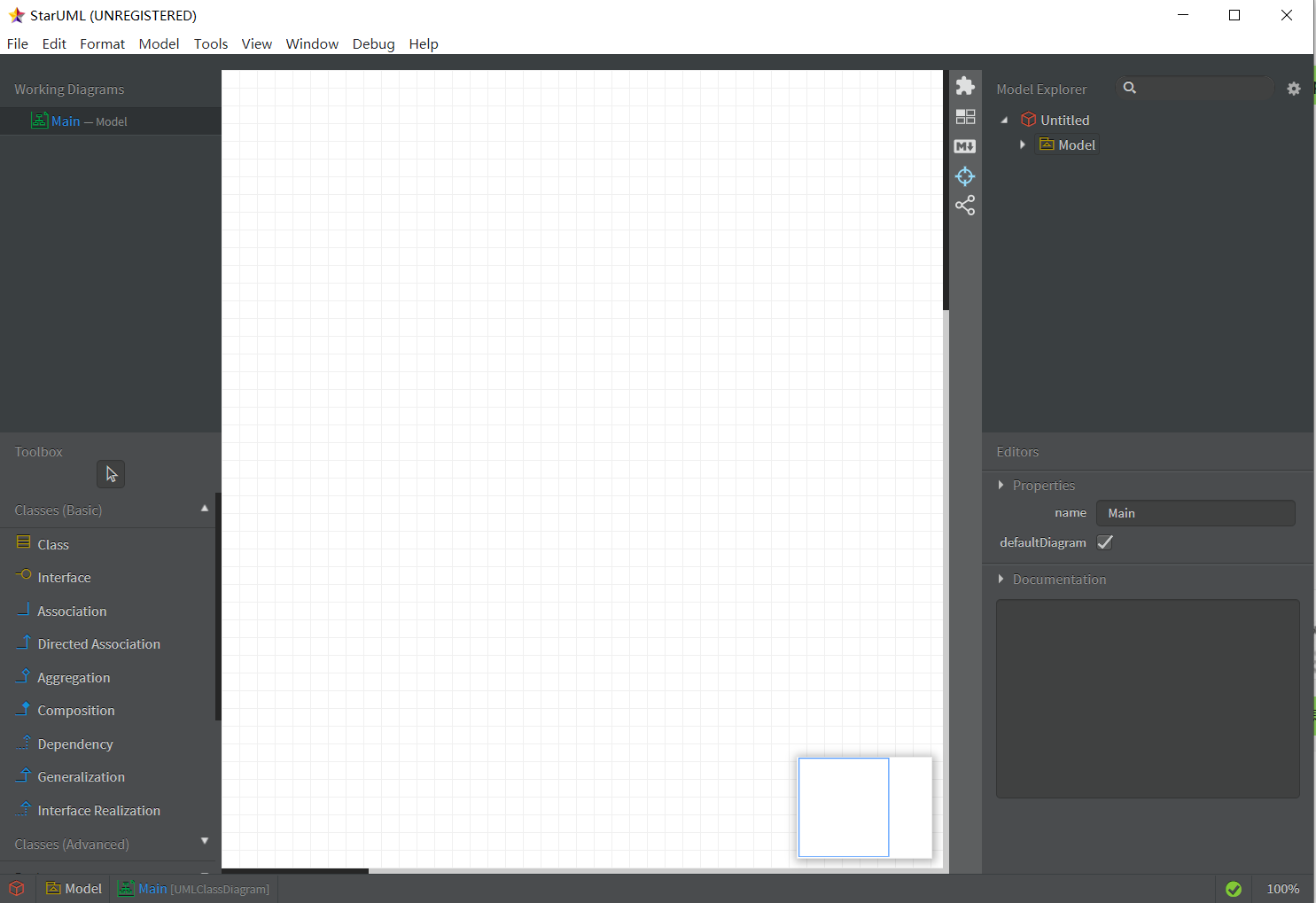
### 3.3.1 StarUML主界面

StarUML的开发界面主要由工具箱、绘图区、模型资源管理器和属性区等构成。

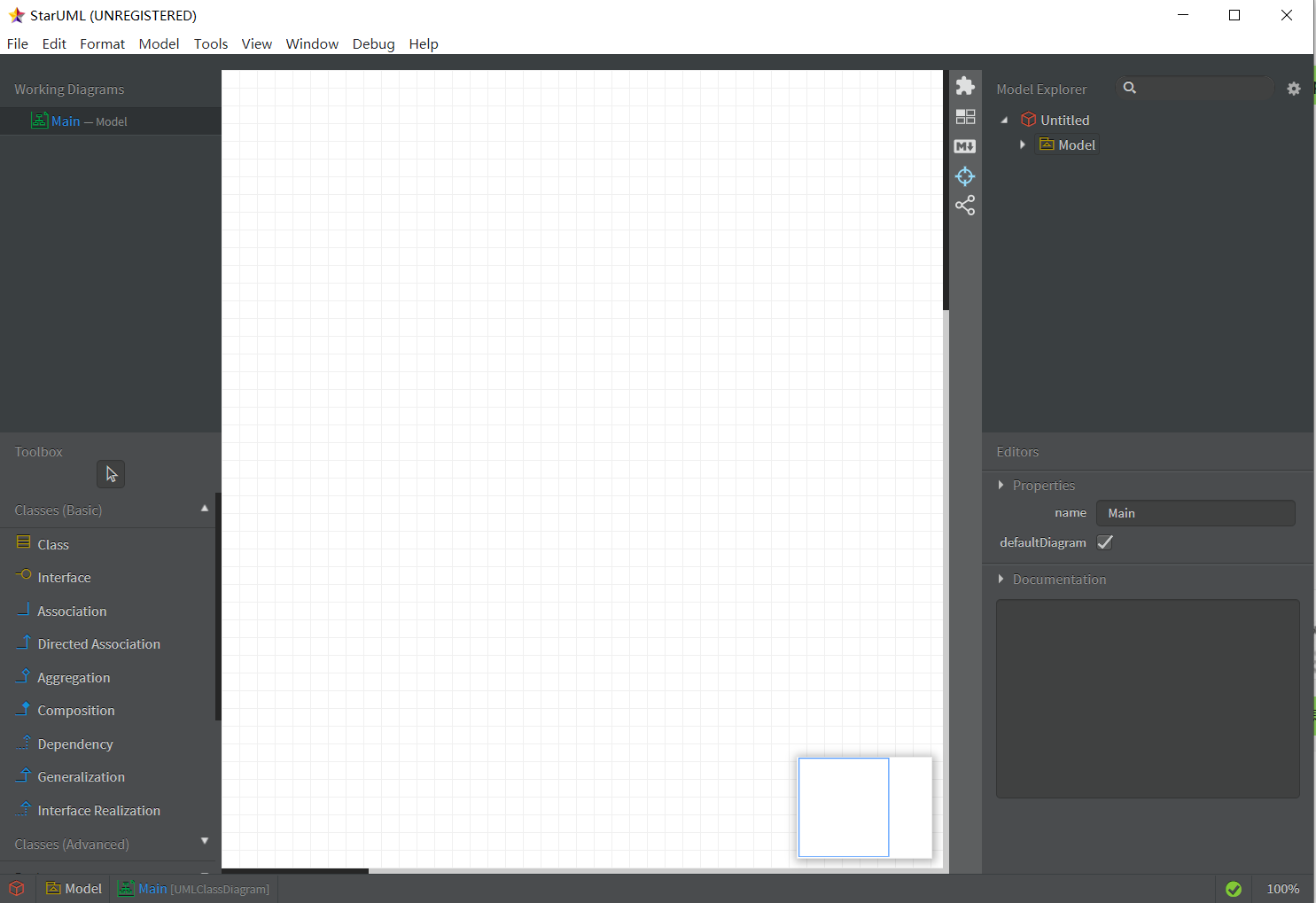
主界面：



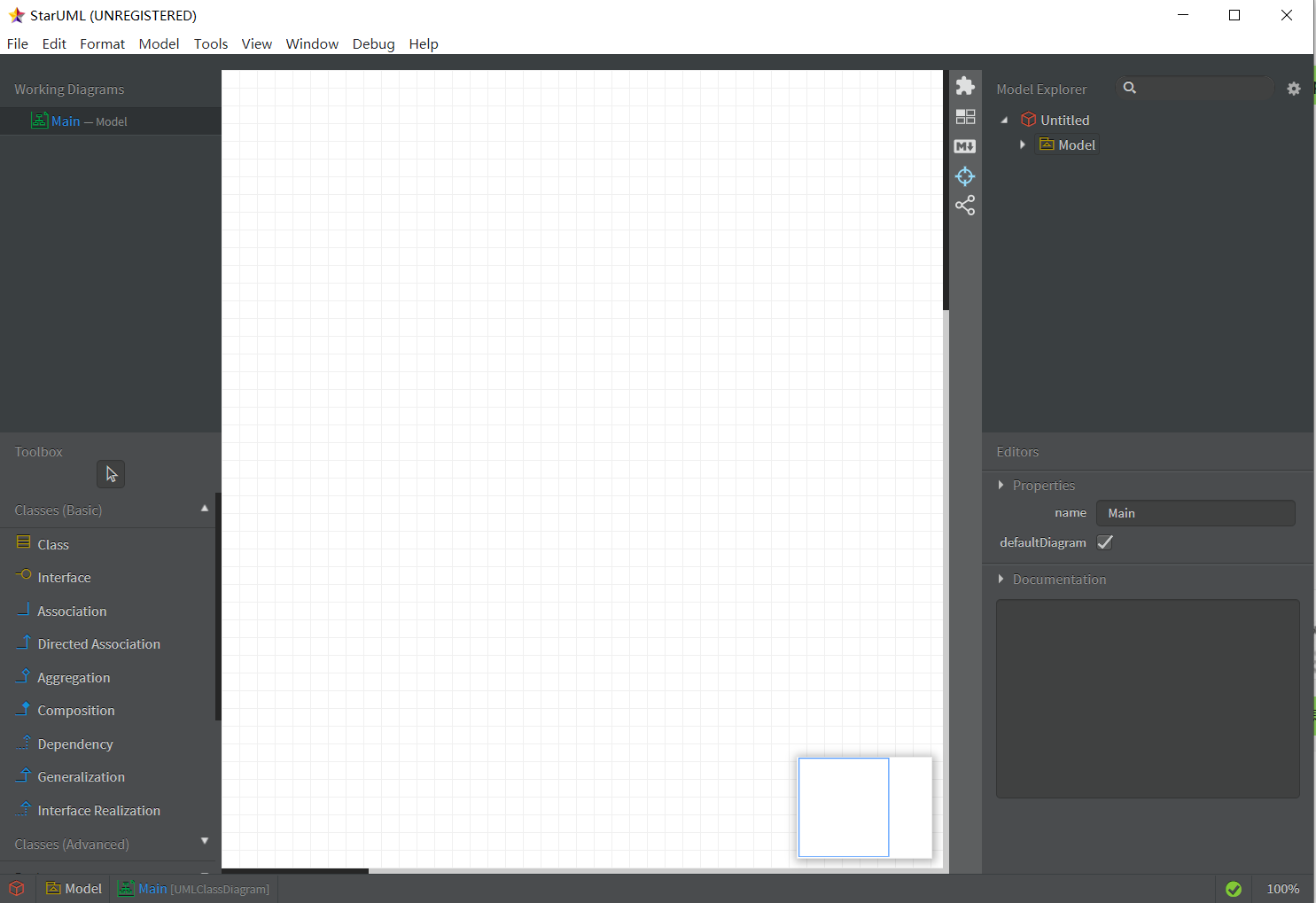
工具区：



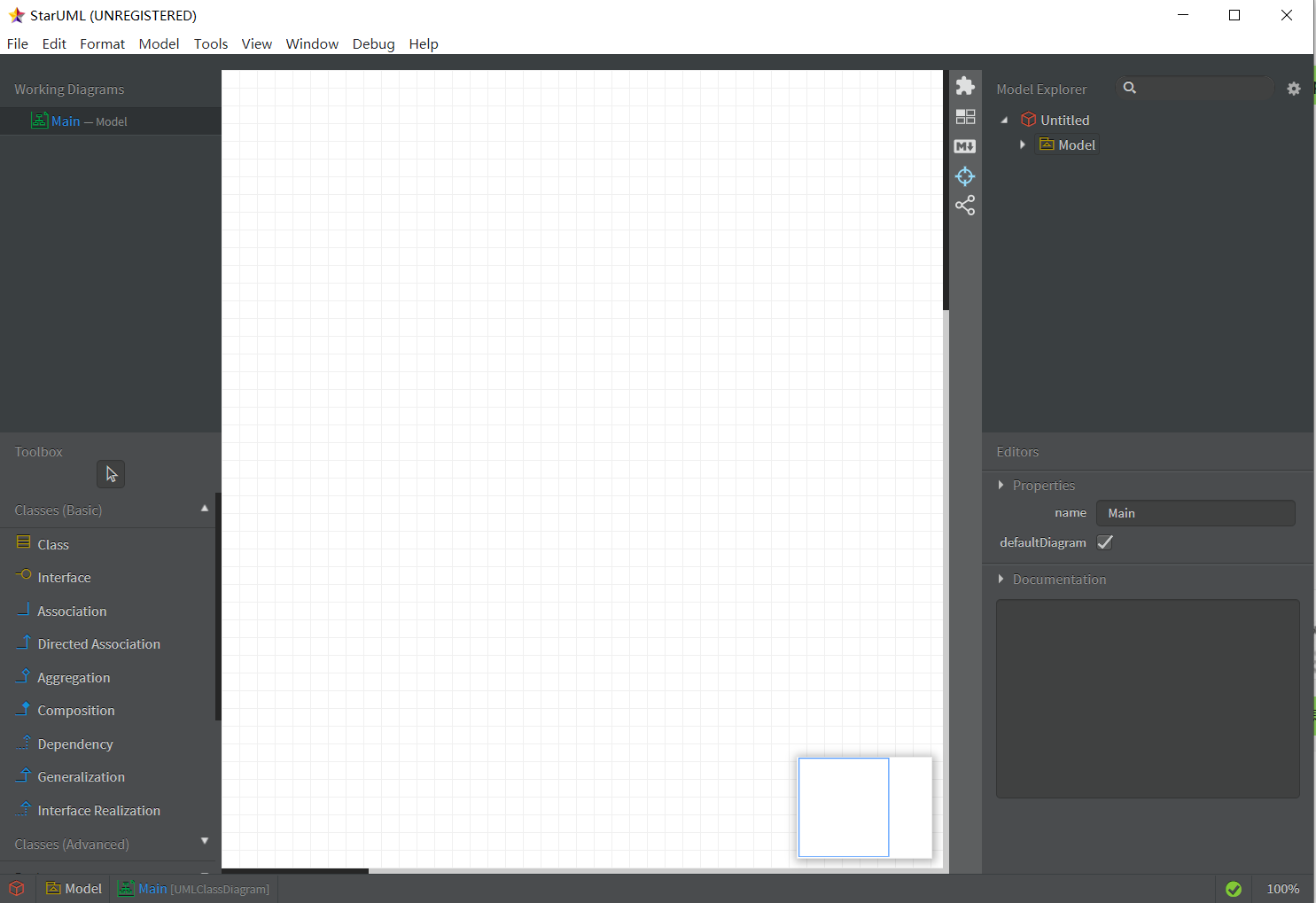
绘图区：



模型资源管理区：



属性区：

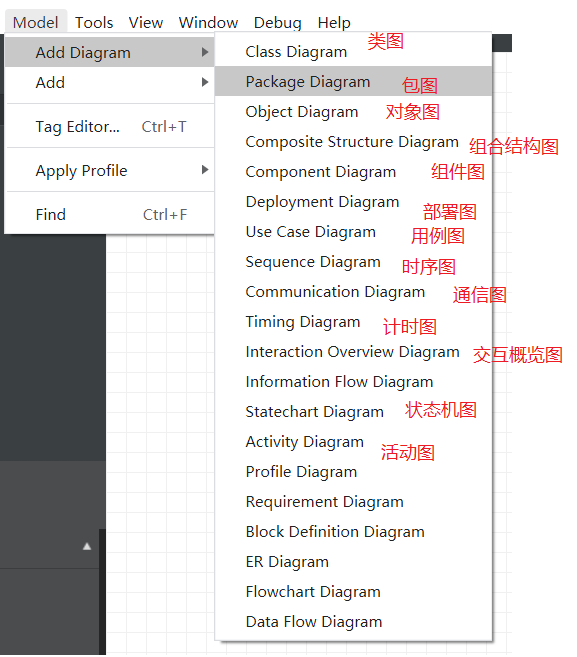


### 3.3.2 StarUML的模型、视与图

StarUML中清晰地区分了模型（Model）、视（View）与图（Diagram）的概念。模型是包含软件模型信息的元素。视则是模型中信息的可视化表达法，图则是表示用户特定设计的可视元素的集合。

UML2.0包含13种图：类图、组合结构图、构件图、部署图、对象图、包图、活动图、顺序图、通信图、交互概览图、计时图、用例图和状态机图。

在StarUML开发中选择Add Diagram操作可以弹出包含13中图的菜单以供设计时选择。



### 3.3.3 StarUML建模的基本过程

1.创建或打开项目（工程）

在StarUML中，项目是基本的管理单位。一个项目可以管理一个或多个软件模型，它是在任何软件模型中都存在的顶级的包。一般地说，一个项目保存在一个文件中。一个项目包含并管理三种子元素。

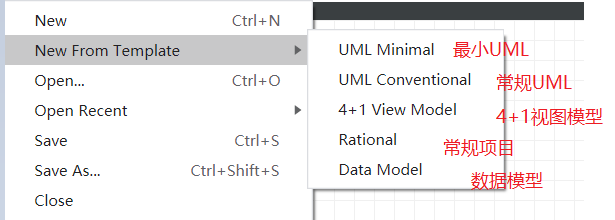
①模型（Model）：管理软件模型的元素

②子系统（Subsystem）：管理表示子系统的模型的元素

③包（Package）：管理元素所需的最一般的元素

（1）创建新项目

启动StarUML或在窗口中选择File→New 或 New From Template选项，选择New from template时，会有多个项目模板跳出。



（2）打开项目

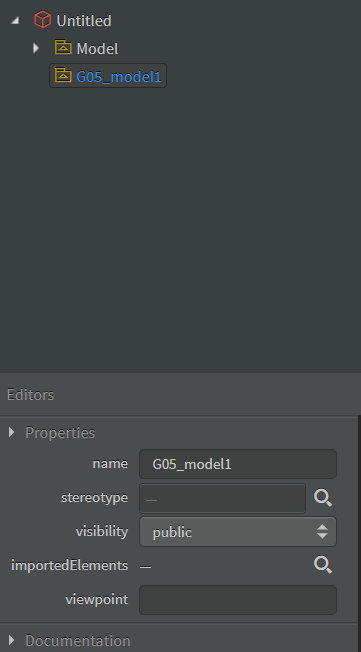
为了继续操作已保存的项目，项目文件必须先打开。

选择File→Open 或Open Recent打开项目框，选择一个项目文件即可打开。

2.创建模块

模块是一种包，它提供了对StarUML功能与特征的扩充。模块的创建可以是几种新扩充元素的结合。不但可以为某用途对一个独立的模块配置扩充元素，而且还可以在同一种模块中创建同一类型的扩充元素。

在StarUML的Model Explorer区域选定Untitled模块，通过右键执行Add Model命令，默认名称为Model1，在其属性区可以修改所需要的名称，如G05\_model1



3.创建参与者和用例

1）创建参与者

参与者定义了在与实体交互时该实体的用户可以发挥作用的一套清楚的角色。参与者可以被认为是对于每个用来交流的用例而言的独立角色。

如果在创建用例图之前创建参与者，则需要经过以下步骤进行。

（1）通过Model主菜单或右击选定模型，选择Add Actor命令；

（2）在模型资源管理器就会出现对应图标；

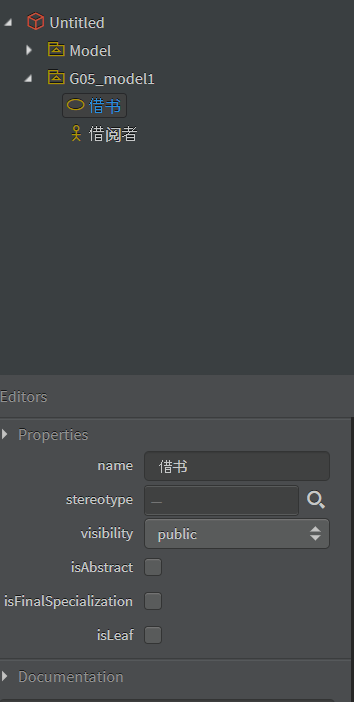
（3）参与者相关属性可以在属性区设置和修改

在用例图中创建参与者，则只需要选择工具箱中的元素即可。

2）创建用例

用例构造用于定义系统行为或其他的语义实体而不展示其内部结构。每个用例指定一系列行为，包括变体，可执行的实体，与参与者实体交互。

其创建过程与参与者类似



4.创建类

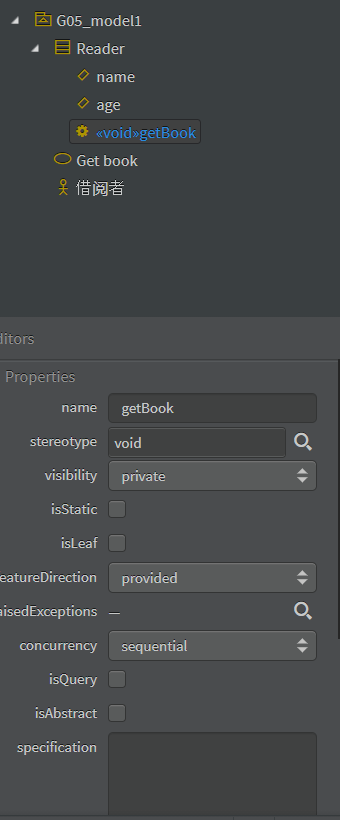
根据需要，同创建参与者一样的方式可以创建类。在创建类时需要为其创建属性及操作，就是Java语言中类中的变量和方法。

1）对类创建属性或操作

在设计图中，右击图中的类目标，在弹出菜单中选择Add Attribute或Operation命令，为类创建属性或操作。

2）对属性或操作设置数据类型

在窗体右下角的属性区，可以在Type输出框输入属性或操作的类型。由于类的封装性和类内部使用的规范性，其内部数据（属性）都是私有的，所以需要把属性设置为Private。



5.创建图

1）创建图

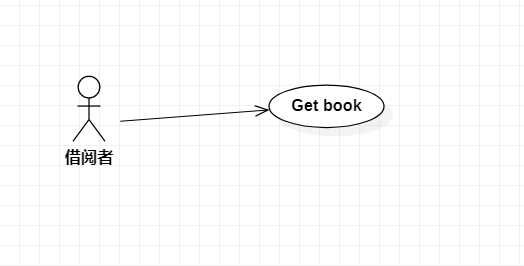
（1）从模型资源管理器选择相应的模型；

（2）右键单击选择Add Diagram，选择相应的图的类型创建图；

2）图中添加元素

（1）从工具箱中选择要创建的元素类型

（2）在图中单击要创建元素的位置，可以通过拖动鼠标区域选择一区域并确定新元素的大小，如果两个元素需要连接到一起，要认真、准确地操作，否则可能会弹出错误提示。

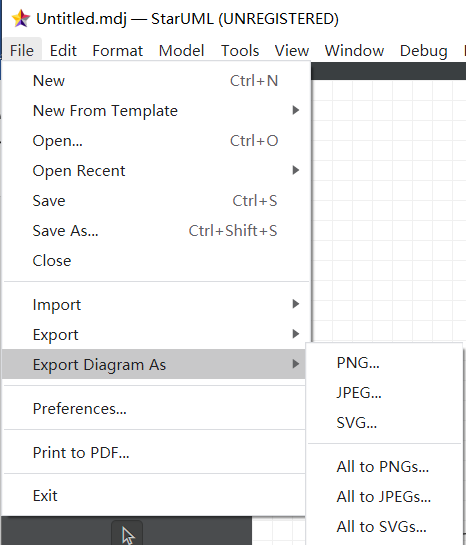


6.保存项目

从File菜单中选择Save命令，所有资料只有一个单一的项目文件（.mdj），所以只有一个文件生产。

7.导出

选择File菜单中Export As Diagram命令可将图表导出，通过选择合适的文件类型保存为其他格式。



# 3.4 双向工程

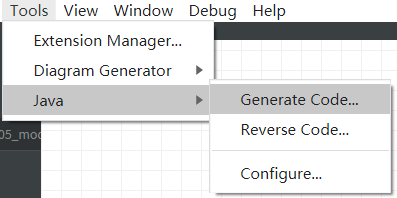
无论是从模型生成代码还是从代码生成模型，都是一项非常复杂的工作。StarUML将正向和逆向工程结合在了一起，并提供了一种在描述系统的架构或设计和代码的模型之间进行双向交换的机制。

### 3.4.1 正向工程

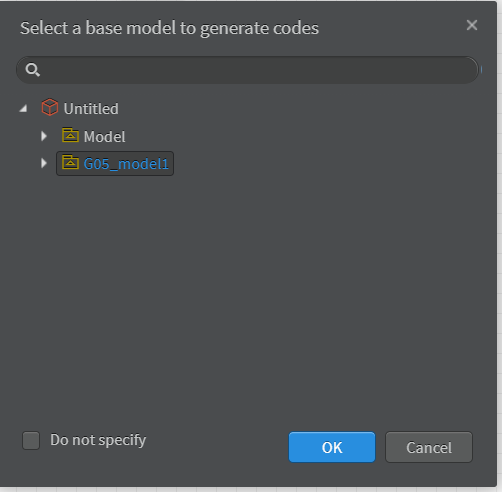
正向工程（代码生成）是指从模型直接生成一个代码框架，这将为程序员节约很多用于编写类、属性、方法代码的琐碎的工作时间。但这不等同于不用编写代码了，而是存在了一个框架，这个框架可以使开发人员思路更加清晰。

生成Java代码的具体步骤如下。

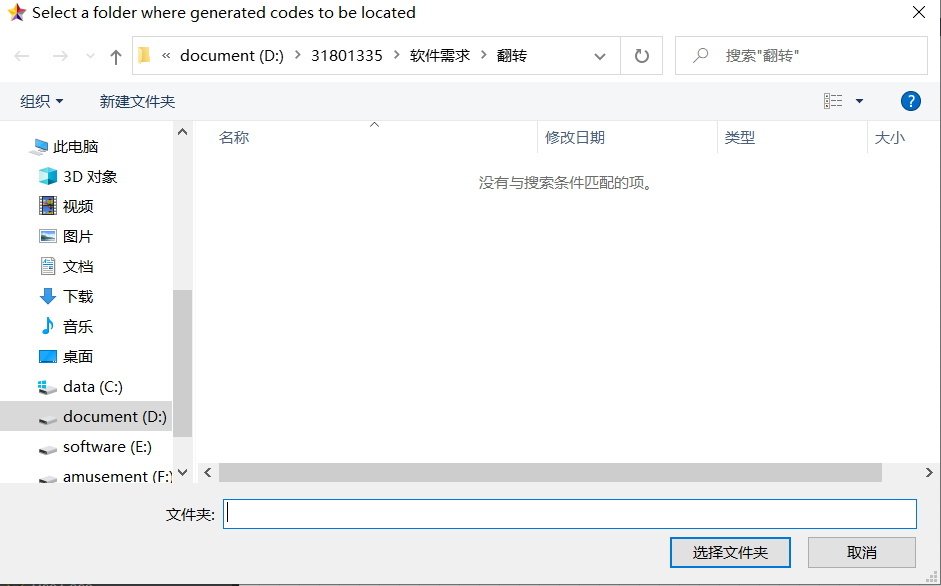
（1）单击Tools→Java菜单，选择Generate Code命令。



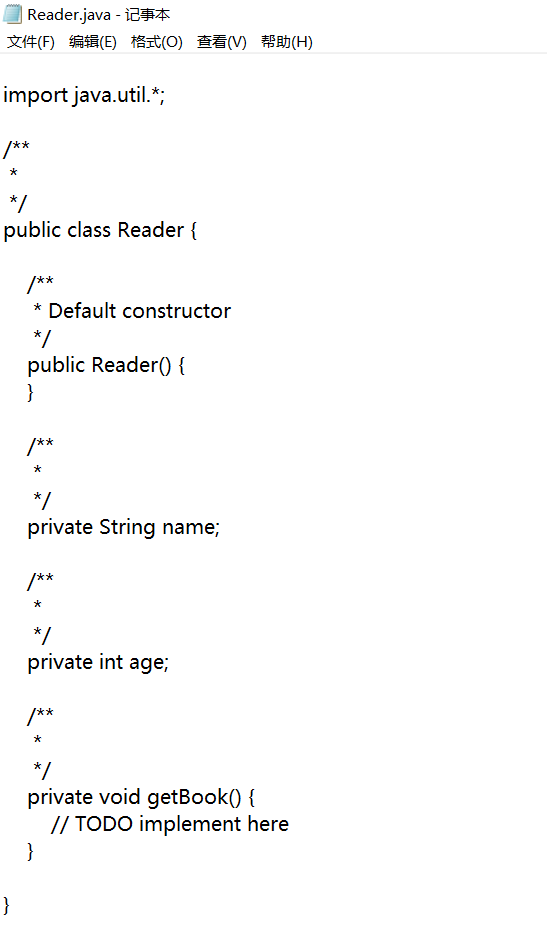
（2）从对话框中选择设计模块



（3）选择有效输出路径



（4）查看输出文件



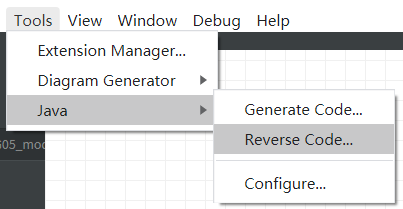
### 3.4.3 逆向工程

逆向工程是分析Java代码，然后将其转化到模型的类的过程。StarUML可以从现有的Java代码创建一个类图，这被称为“reverse engineering“，当从现有的代码生成图表，或者修改了生成的代码，并且想在图表中反映出来时，就要启用逆向工程。

通过图表或者文本编辑器去反复工作的过程，是面向对象编程中的一个基本过程，被称为round-trip engineering。

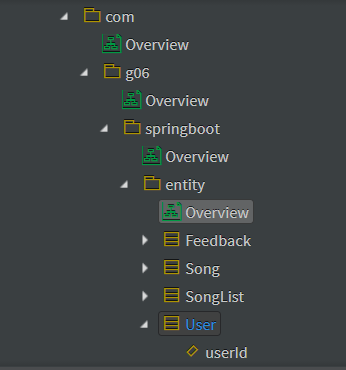
逆向工程实现步骤如下：

（1）选择Tools→Java，选择Reverse Code

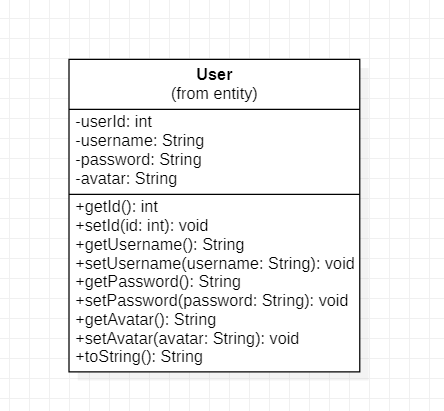


（2）选择Java代码所在目录或文件

（3）生成模型



（4）查看类图



# 问题：

（1）简述题：在StarUML中包含的几种元素是什么，分别有什么作用？

答案：一个项目包含并管理三种子元素。①模型（Model）：管理软件模型的元素。②子系统（Subsystem）：管理表示子系统的模型的元素。③包（Package）：管理元素所需的最一般的元素。

（2）判断题：正向工程是通过到实现语言的映射而把模型转化为代码的过程，这将为程序员节约很多用于编写类、属性、方法代码的琐碎的工作时间。

答案：错，正向工程是指从模型直接产生一个代码框架，是代码生成的过程。而逆向工程是通过到实现语言的映射而把模型转化为代码的过程。